Examen de Java : ISIMA : 2ème année, Filière 2 et 3
Durée 2h, tous documents autorisés.

Question 1 (4 points) :

Soit une classe Emprunt : déclarer trois attributs « private », montant du prêt, taux d'intérêt, et nombre d'années pour l'emprunt. (1 pt). Déclarer un constructeur qui accepte des valeurs pour ces trois attributs (1 pt). Déclarer les méthodes d'accès pour les 3 attributs (1 pt). Déclarer une méthode paiement, qui retourne le montant d'un paiement mensuel (simplifier le calcul en considérant un paiement à taux fixe) (1 pt).

Questions 2 (5 Points) :

a) Comment rappeler l'exécution d'une méthode de la superclasse au sein de la redéfinition d'un méthode de même nom ? (Codez par exemple dans une sous-classe une méthode finalize qui fait appel au finalize de sa superclasse) (1 pt)
b) Lequel de ces mots n'est pas un mot clé réservé en Java : if, then, goto, case (1 pt)
c) Quel est le résultat de la compilation et de l'exécution du code suivant :

```
public class Conv{
      public static void main(String argv[]){
            Conv c=new Conv();
            String s=new String("ello");
            c.amethod(s);
      }
      public void amethod(String s){
            char c='H';
            c+=s;
            System.out.println(c);
      }
}
```

Réponse : (1) Hello (2) ello (3) elloH (4) Erreur à la compilation

d) Avec le code suivant, comment peux-t-on faire pour invoquer le constructeur de la classe Base qui affiche la chaîne de caractères "base constructor" :

```
class Base{
      Base(int i){
            System.out.println("base constructor");
      }
      Base(){
      }
}
public class Sup extends Base{
      public static void main(String argv[]){
            Sup s= new Sup();
            //One
      }

      Sup() {
            //Two
      }
      public void derived() {
            //Three
      }
}
```

Réponses:

(1) En remplaçant //One par Base(10); (2) En remplaçant //One par super(10);
(3) En remplaçant //Two par super(10); (4) En remplaçant //Three par super(10);

e) Quel est le résultat de la compilation et de l'exécution du code suivant :

```
public class Pass{
        static int j=20;
        public static void main(String argv[]){
              int i=10;
              Pass p = new Pass();
              p.amethod(i);
              System.out.println(i);
              System.out.println(j);
        }
        public void amethod(int x){
              x=x*2;
              j=j*2;
        }
  }
```

Réponse : (1) Erreur : a method parameter does not match variable

(2) 20
40

(3) 10
40

(4) 10
20

Question 3 (2 points) :

Le compilateur crée automatiquement un nouvel objet String à chaque chaîne de caractère littérale rencontrée.

a) `String s = "Hola Mundo";`

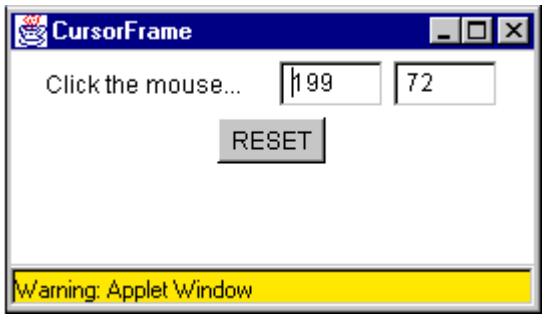
Le code ci-dessus est équivalent à celui donné ci-dessous. A votre avis quel est le code le plus efficace (1 pt) et combien d'objets String sont créés dans le cas a) et dans le cas b) (1 pt).

b) `String s = new String("Hola Mundo");`

NOM :	PRENOM :	FILIERE :

Question 4 : Analyse de code (commenter sur la feuille d'examen)

A) Sur 5 points : commentez le code source suivant afin de montrer votre connaissance du langage et votre maîtrise de ce que vous avez étudié en TP.




```
import java.awt.*;
import java.awt.event.*;

public class CursorFrame extends Frame {
   TextField a, b;
   Button    btn;

   public CursorFrame() {
      super("CursorFrame");
      setSize(400, 200);
      setLayout(new FlowLayout());
      add(new Label("Click the mouse..."));
      a = new TextField("0", 4);
      b = new TextField("0", 4);
      btn = new Button("RESET");
      add(a); add(b); add(btn);

      addMouseListener(new MouseAdapter() {
         public void mousePressed(MouseEvent e) {
            a.setText(String.valueOf(e.getX()));
            b.setText(String.valueOf(e.getY()));
         }
      });

      addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent e) {
            setVisible(false);
            dispose();
            System.exit(0);
         }
      });

      btn.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            a.setText("0");
            b.setText("0");
         }
      });
  }

  public static void main(String[] args) {
     CursorFrame app = new CursorFrame();
     app.setVisible(true);
     }
  }
```

NOM :	PRENOM :	FILIERE :

B) Sur 4 points : Lors d'un entretien d'embauche on vous demande d'expliquer le comportement des codes C++ et Java suivants, en fonction de vos connaissances en Java, C et C++. Voici les informations dont vous disposez (les résultats sont donnés, il faut expliquer ce qui se passe):

```
// code C++ (class string STL)
string * s1 = new string("hello");
string * s2 = s1;

(*s1) += " world";
cout << *s1 << endl << *s2 << endl;

// Resulat
// s1 = s2 = "hello world"
```

```
// Code Java

String s1 = "hello";
String s2 = s1;
s1 += " world";

System.out.println(s1 + "\n" + s2);

// Resultat s1 = "hello world"
// et s2 = "hello"
```

Le code Java ci-dessous, grâce à la méthode `identityHashCode` de la class System, donne "l'adresse" des variables dans la machine virtuelle Java.

```
// Code Java
String s1 = "hello";
String s2 = s1;
System.out.println("s1 = " + s1 + "; s2 = " + s2);
System.out.println("System.identityHashCode(s1) = " + System.identityHashCode(s1));
System.out.println("System.identityHashCode(s2) = " + System.identityHashCode(s2));

s1 += " world";
System.out.println("\ns1 = " + s1 + "; s2 = " + s2);
System.out.println("System.identityHashCode(s2) = " + System.identityHashCode(s2));
```

L'exécution du code ci-dessus donne le résultat suivant :

```
s1 = hello; s2 = hello
System.identityHashCode(s1) = 2452092
System.identityHashCode(s2) = 2452092

s1 = hello world; s2 = hello
System.identityHashCode(s1) = 7474923
System.identityHashCode(s2) = 2452092
```

Because the compiler automatically creates a new String object for every literal string it encounters, you can use a literal string to initialize a String.

```
String s = "Hola Mundo";
```

The above construct is equivalent to, but more efficient than, this one, which ends up creating two Strings instead of one:

```
String s = new String("Hola Mundo");
```

The compiler creates the first string when it encounters the literal string "Hola Mundo!", and the second one when it encounters new String

---

Note to C and C++ Programmers: The shortcut assignment operator += when used with Strings may confuse C and C++ programmers at first. Recall that a += b is equivalent to a = a + b. Let's look at two code samples written in C++ and the Java programming language:

```
//C++ code
string*  s1 = new string("hello");
string*  s2 = s1;
(*s1) += " world";
cout<<*s1<<endl<<*s2<<endl;
return  0;
//s1 = s2 = "hello world"
```

```
//Java progamming language code
String s1 = "hello";
String s2 = s1;
s1 += " world";
System.out.println(s1 + "\n" + s2);
//s1 = "hello world" and  s2 = "hello"
```

In the C++ example, the strings s1 and s2 print the same result because they both point to the same address. In the Java programming language, Strings can't be modified, so the + operator must create a new String when " world" is appended to s1.

The following code sample illustrates that s1 and s2 point to the same object until you use the += operator to assign a new String to s1.

```
//Java programming language code
String s1 = "hello";
String s2 = s1;
System.out.println("s1 = " + s1
                   + "; s2 = " + s2);
System.out.println("System.identityHashCode(s1) = "
                   + System.identityHashCode(s1));
System.out.println("System.identityHashCode(s2) = "
                   + System.identityHashCode(s2));

s1 += " world";
System.out.println("\ns1 = " + s1
                   + "; s2 = " + s2);
System.out.println("System.identityHashCode(s1) = "
                   + System.identityHashCode(s1));
System.out.println("System.identityHashCode(s2) = "
                   + System.identityHashCode(s2));
```

Here's the output:

```
s1 = hello; s2 = hello
System.identityHashCode(s1) = 2452092
System.identityHashCode(s2) = 2452092

s1 = hello world; s2 = hello
System.identityHashCode(s1) = 7474923
System.identityHashCode(s2) = 2452092
```

s1 points to a new address after " world" is appended.

```
import java.lang.Math;
import java.io.DataInputStream;

class loan {
        private float principal = 0;
        private float interestRate = 0;
        private int numYears = 0;

        loan(float thePrincipal, float theInterest,
                int theYearsToRepay)
        {
                principal = thePrincipal;
                interestRate = theInterest;
                numYears = theYearsToRepay;
        }

        // accessor methods
        float getPrincipal()
        {
                return principal;
        }

        float getInterestRate()
        {
                return interestRate;
        }

        int getYears()
        {
                return numYears;
        }

        double calculatePayment()
        {
                // assuming monthly payments
                double interest = interestRate/100;
                double payment = (interest * principal/12) /
                                 (1 - Math.pow(interest/12 + 1,
                                               -12*numYears));
                return payment;
        }
}
```